ADL Hw2 Report
BERT for QA

B06902104 吳由由

1 Tokenization

1.1 How BERT tokenizer works

Bert先使用BasicTokenizer依照文章輸入的空格進行一般分詞，在簡單分詞後，再送入WordpieceTokenizer將前面的結果切成更小單位的wordpieces. wordpieces的功能是幫忙處理OOV,讓一個不在字典裡面的詞由word piece組成，如此可以讓[UNK]的量減少

- BasicTokenizer
  - 將輸入轉成unicode
  - 將無意義的字元去除(control, whitespace)
  - 判斷該unicode是否是中文字，如果是的話就以中文字一個一個字分詞
  - 如果不是中文，轉成小寫並normalize text後斷詞
  - 以標點符號斷開
  - 切成wordpieces

1.2 Observation on the method on different strings

對於英文字或數字，wordpiece tokenizer可以把詞再切成更小的wordpiece來避免OOV,例如所有數字的組合有無限多種，字典不可能把所有數字都存進字典裡，所以把一個數字切成幾個短數字拼起來就能解決這個問題。但在中文上，每個字元已經是最小單位了不能再被分割，因此中文上沒有影響，不在字典裡的中文字會直接用[UNK]表示。
Example:

- 1275萬→['127', '##5', '萬']
- unfortunately →['u', '##n', '##fo', '##rt', '##un', '##ate', '##ly']

另外，因為我們使用的是bert-base-chinese model裡面所含的英文單字量比較少，所以一般常見的英文單字也會被切開成很多word pieces

2 Answer Span Processing

2.1 The way to convert the answer span start/end position

- 在training data中已經有給answer_start的character位置，我把從開頭到answer_start的context拿去bert tokenizer裡分詞，得到被字被tokenize後的list，這個list的長度就是answer start token的開始長度
- training data中有給answer的text, 拿去tokenizer裡tokenize後量長度加上start position即可得到end position

2.2 The rule to determine start/end position

- 限制end position > start position
- 限制end position − start position > 30
- 當以上rule被違反時，保留end/start機率比較大的位置，比較小那端則找那端機率下一大的位置來算有無違反上面兩條rule，這個process只維持最多五次，若五次還是找不到合適的答案區間則把這題當作unanswerable

3 Padding and Truncating

3.1 Maximum input token length for bert-base-chinese

The maximum input token length of bert-base-chinese is 512

3.2 The method for pad or truncate

限制question的長度最長為50, 若問題長度超過50, 因為我觀察問題的重點幾乎都集中在開頭跟結尾，我取問題的前30個token跟最後20個token組成新的question. 剩下$512 - length(question) - 3$(for special token)給context, 若context的長度超過剩餘空間，則把context從後面truncate, 如果truncate掉的部份含有answer則把這筆資料改成unanswerable。以整個batch裡最長的資料作為這筆batch的length, 比batch length短的data在後面pad 0

4 Model

4.1 How model predict if the question is answerable

取出Bert model預測的結果pooled output, 將pooled output接上一層dropout再接Linear以預測這個question是否為answerable

$$\begin{aligned} output &= bert(context, attention, token_type_id) \\ output &= output[1] \text{ //get pooled output} \\ answerable &= Linear(Dropout(output)) \end{aligned}$$

4.2 How model predict the answer span

取出Bert model對每個token位置預測出的output(batchsize, sequence length)，過一層linear作為start/end該出現在哪個位置的機率預測(512個位置視為分成512類)

$$\begin{aligned} output &= bert(context, attention, token_type_id) \\ output &= output[0] \text{ //get hidden output} \\ start &= Linear(output) \\ end &= Linear(output) \end{aligned}$$

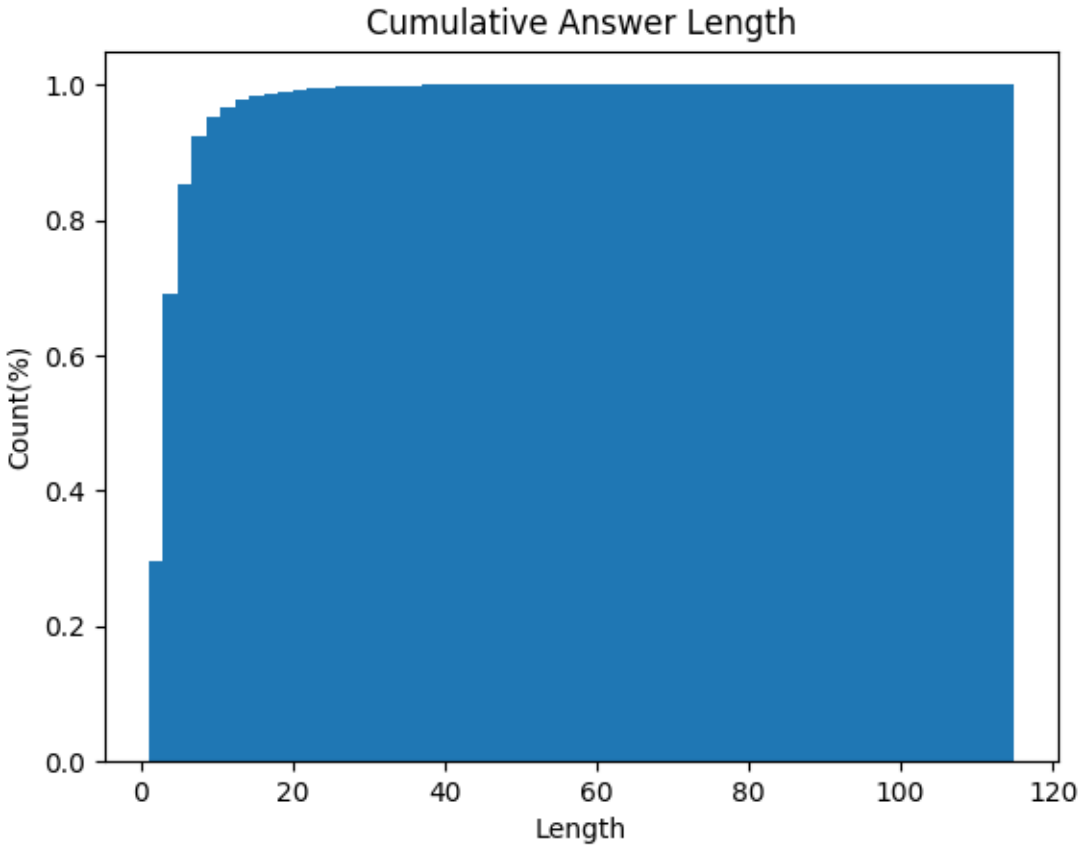
4.3 Loss function

- For answerable/unanswerable task: BCEWithLogitsLoss with positive weight 0.48.
- For answer span task: CrossEntropyLoss, ignore index = -1, if the question is unanswerable, the start/end label is -1.

4.4 Optimization algorithm

- AdamW, which is implements adam algorithm with weight decay fix. Learning rate is 2e-5, eps is 1e-8.

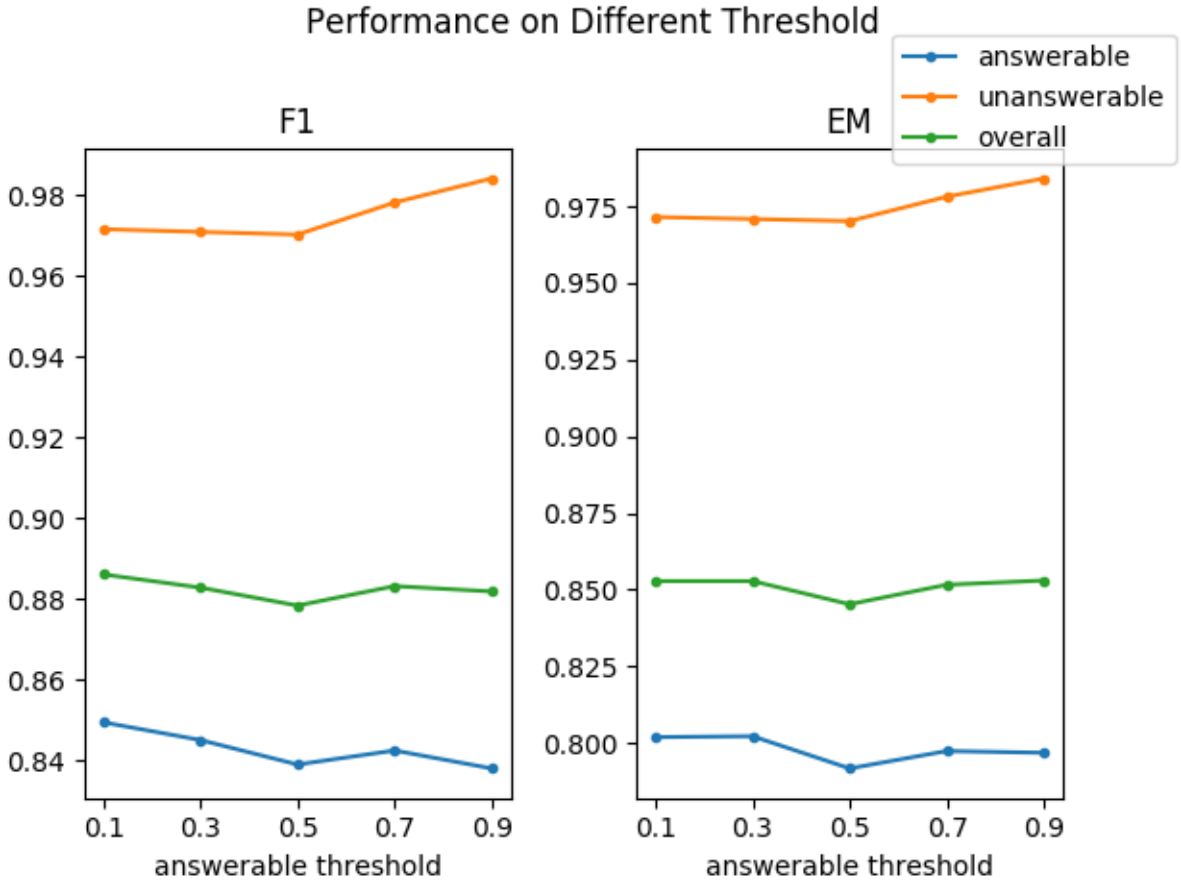
5 Answer Length Distribution



由culmulative distribution of answer length的圖可以觀察到在training set上的answer length大多不超過30, 因此在predict的時候若end-start超過30的答案可以直接去除掉，因為他們less likely是正確答案

6 Answerable Threshold

The probability threshold I use is 0.5



7 Extractive Summarization

- Bert可以幫助找出良好的sentence embedding，讓bert讀文章後得到對應的sentence output embedding（可實驗bert不同層的output得到不同embedding)，並使用K-means演算法把sentence依照語意分群，語意相似的會聚在一起，每群找到centroid後找離centroid最近的句子作為extractive summary.
- Reference: https://arxiv.org/pdf/1906.04165.pdf